(2)

Technical Memorandum TRAC-F-TD-0290
January 1990

AD-A228 808

A TOP ATTACK FIRE METHODOLOGY

U. S. ARMY
TRADOC ANALYSIS COMMAND
OPERATIONS ANALYSIS CENTER
MODEL DEVELOPMENT & MAINTENANCE DIRECTORATE
FORT LEAVENWORTH, KS 66027-5200

Technical Memorandum TRAC-F-TD-0290
JANUARY 1990

TRADOC Analysis Command-Fort Leavenworth (TRAC-FLVN)
Model Support Directorate
Fort Leavenworth, Kansas 66027-5200

A TOP ATTACK FIRE METHODOLOGY

by

Marvin R. Methven

ACN 16306

CERTIFIED BY:

H. KENT PICKETT
Director, Model Support Directorate

APPROVED BY:

ROBERT LA ROCQUE
Director, TRAC-FLVN

Distribution Statement:
Approved for public release;
distribution is unlimited

9100623

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release
Distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Technical Memorandum TRAC-F-TD-0290

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
TRAC-Fort Leavenworth

6b. OFFICE SYMBOL *(If applicable)*
ATRC-FM

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Ft Leavenworth, KS 66027-5200

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
TOP ATTACK

12. PERSONAL AUTHOR(S)
Mr. Marvin Methven

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
January 1990

15. PAGE COUNT
42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report describes a methodology put into a preprocessor program for computer war-game models. The preprocessor computes kill rates or probability of kills (Pk) for selected direct-fire weapons against selected targets. These are input data used by the War-game models.

The methodology is an expansion to a methodology already in the preprocessor. This methodology computes probability of his (Ph) for the weapon fired against the target. Ph is an intermediate value used in computing the kill rate or Pk.

The preprocessor needed the expansion to accommodate a new type of weapon which has a flight path above the target. The previous methodology assumed a ground-level flight path.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Mr. H. Kent Pickett

22b. TELEPHONE *(Include Area Code)*
(913)684-4510

22c. OFFICE SYMBOL
ATRC-FM

DD Form 1473, JUN 86 — *Previous editions are obsolete.* — SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

FIGURES

ABSTRACT

This report describes a methodology put into a preprocessor program for computer war-game models. The preprocessor computes kill rates or probability of kills (Pk) for selected direct-fire weapons against selected targets. These are input data used by the war-game models.

The methodology is an expansion to a methodology already in the preprocessor. This methodology computes probability of hit (Ph) for the weapon fired against the target. Ph is an intermediate value used in computing the kill rate or Pk.

The preprocessor needed the expansion to accommodate a new type of weapon which has a flight path above the target. The previous methodology assumed a ground-level flight path.

1. Introduction. A White Sands Missile Range (WSMR) program called Direct-Fire Weapon Preprocessor (DFWP) computes kill rates for Vector-In-Commander (VIC) and other computer war-game models. The kill rate data are for selected direct-fire weapons against selected targets. Because of new top-attack weapons, the methodology in this program was not adequate. This report discusses the problem and defines the solution.

2. Problem. The methodology for probability of hit (Ph) needed to be changed.

a. Overview of Ph computation.

(1) DFWP uses two kinds of data.

(a) One is the target dimensions. This is available from the Ballistic Research Laboratory (BRL).

(b) The other is the probability density data for round impacts of the weapon. This is available from the Army Materiel Systems Analysis Activity (AMSAA).

(2) The Ph methodology has two parts.

(a) DFWP computes target area from the target dimensions and weapon flight path. The orientation of the target (aspect angle) and vertical angle of weapon approach (attack angle) determine the flight path.

(b) DFWP computes Ph by integrating the probability density function (pdf) of round impact p(X,Y) over the target area.

$$Ph = \iint_{\text{Target area}} p(X,Y)\, dY\, dX$$

b. Problem.

(1) DFWP must compute target area differently for top-attack weapons than for the other weapons. Those weapons have an attack angle of zero degrees to the ground. The target area computes to be a rectangle or rectangles. Target area for top-attack weapons must be computed by an expanded methodology, and it computes to be an irregular polygon.

(2) DFWP must compute Ph differently for top-attack weapons. DFWP computes PH for the other weapons (paragraph 2a(2)(b)) by table lookup. This is because the target area is a rectangle or two rectangles. DFWP must compute Ph for top-attack weapons by a numerical method since the target area is more complicated.

3. Assumptions.

a. The DFWP assumes the targets to be a box or two boxes, one on top of the other. If two boxes are assumed, then the bottom box is called the hull, and the top box is called the turret. The turret is assumed to be exactly on the center of the hull (see figure 1 (a) and (c)). (This last assumption might change in the future.)

b. The DFWP assumes p(X,Y) to be normal and independent in the horizontal and vertical directions.

c. The DFWP computes both fully exposed and defilade targets. Defilade targets are behind a barrier of sorts and are partially hidden from view. The distance of the barrier to the target in addition to the barrier height determines the exposed part of the target for top-attack weapons. Barrier height is an input to DFWP but not barrier distance. DFWP assumes the barrier distance equal to the hull height.

d. The DFWP assumes aimpoint to be the center of target volume.

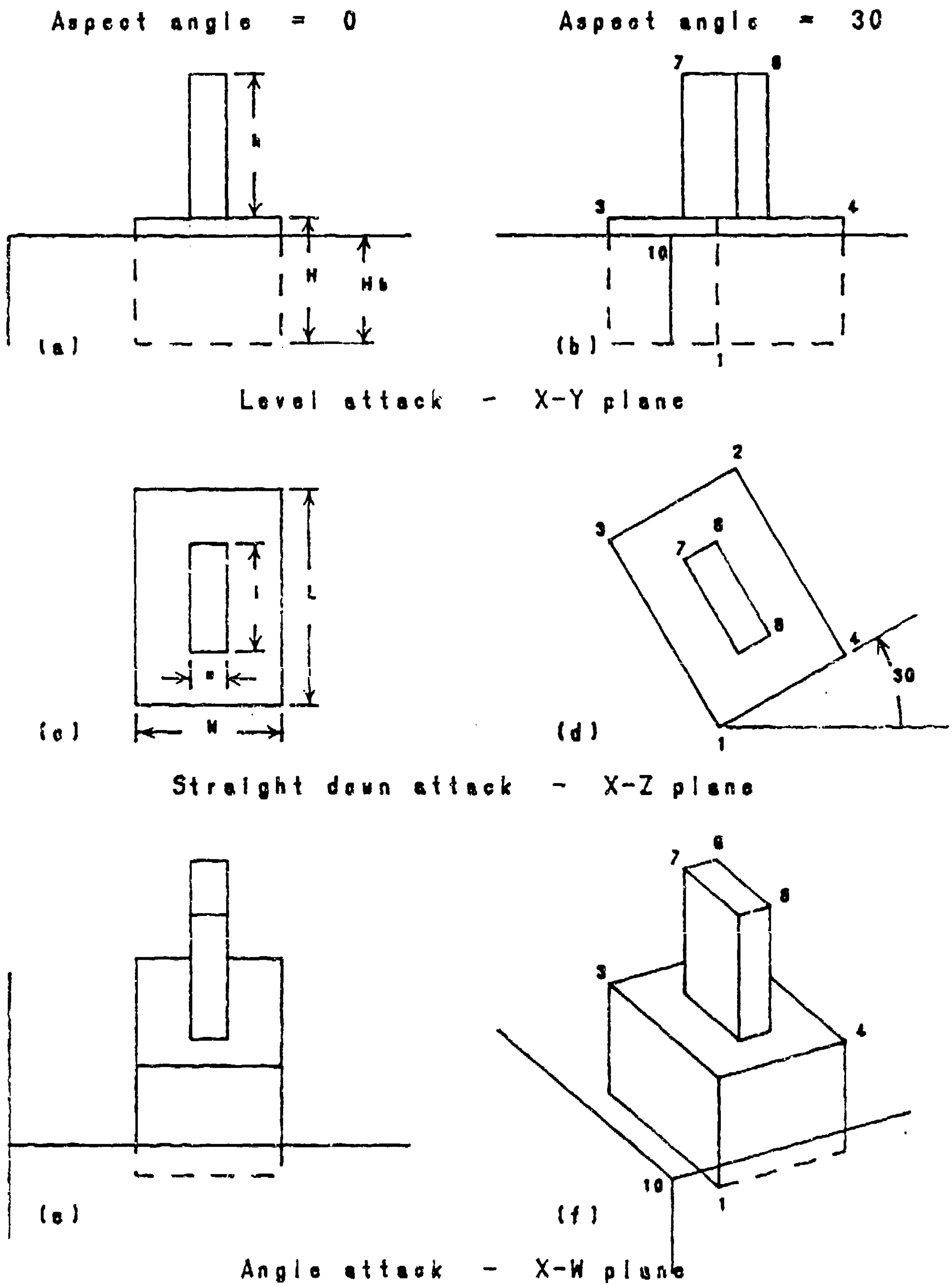

Figure 1. Target areas

4. Target area.

a. General.

(1) Summary. The DFWP computes target area from the weapon flight path and the target dimensions. The aspect angle and attack angle determine the weapon flight path. DFWP uses a coordinate system for the target dimensions.

(2) Aspect angle. The DFWP computes Ph for the weapon engaging the target from the front, side, and other directions. This direction is the aspect angle. Engagement from the front is a zero-degree aspect angle. Engagement from the side is a 90-degree aspect angle.

(3) Attack angle. Each weapon has an attack angle. Paragraph b discusses weapons with an attack angle of zero degrees. This was the only case considered previously, and paragraph b discusses the methodology that was already in the DFWP. Paragraph c discusses weapons with an attack angle of 90 degrees. Paragraph d discusses weapons with an attack angle between 0 and 90 degrees.

(4) Coordinate system. The corner coordinates describe the target and barrier. The corners are:
(See figure 1 (b), (d), and (f).)

Hull	Turret	Barrier
1 - Front bottom		10 - Front top
2 - Back top	6 - Back top	
3 - Left top	7 - Left top	
4 - Right top	8 - Right top	

The coordinate system has the X-axis for the horizontal direction, the Y-axis for the vertical direction, the Z-axis for the depth direction, and the origin, O, at the center top of the hull. The target is rotated abouc the origin for the different aspect angles. The coordinates for point 1 are X1, Y1, Z1, and W1; the coordinates for point 2 are X2, Y2, Z2, and W2; etc. The X and Y coordinates describe the target area for level attack. The X and Z coordinates describe the target area for straight down attack. X and W coordinates describe the target area for angle attack. The W-axis is an additional axis made perpendicular to the round flight path and to the X-axis.

(5) Figure 1 shows target areas. The dimensions are:

W = Hull width = 2 meters (m) w = Turret width = .5 m
L = Hull length = 3 m l = Turret length = 1.5 m
H = Hull height = 1.75 m h = Turret height = 2 m
Hb = Barrier height = 1.5 m
A = Aspect angle = 30 degrees

b. Level attack.

(1) Target area formulas.

(a) Coordinates for hull corners 3 and 4.

$$X4 = W/2 * \cos(A) + L/2 * \sin(A)$$
$$X3 = -X4$$
$$Y3 = Y4 = 0$$

(b) Coordinates for turret corners 7 and 8.

$$X8 = w/2 * \cos(A) + l/2 * \sin(A)$$
$$X7 = -X8$$
$$Y7 = Y8 = h$$

(c) Coordinates for barrier corner 10.

$$X10 = -(W/2 + H) * \cos(A) + (L/2 + H) * \sin(A)$$
$$Y10 = -H + Hb$$

(d) Coordinates for aimpoint.

$$X0 = 0$$
$$Y0 = (h + Y10) / 2$$

(2) Ph formula. DFWP integrates pdf p(X,Y) over the target area of hull and turret.

$$Ph = \text{Ph on hull} + \text{Ph on turret}$$

$$Ph = \int_{X3}^{X4}\int_{Y10}^{0} p(X,Y)\, dY\, dX + \int_{X7}^{X8}\int_{0}^{h} p(X,Y)\, dY\, dX$$

c. Straight-down attack.

(1) Target area formulas.

(a) Hull coordinates.

Let: A1 = A = aspect angle.
A2 = A1 + 90.
M = rotation matrix.

$$M = \begin{vmatrix} \cos(A1) & -\sin(A1) \\ \sin(A1) & \cos(A1) \end{vmatrix}$$

Then:

$$\begin{vmatrix} X1 \\ Z1 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ -L/2 \end{vmatrix} \qquad \begin{vmatrix} X2 \\ Z2 \end{vmatrix} = M \begin{vmatrix} W/2 \\ L/2 \end{vmatrix}$$

$$\begin{vmatrix} X3 \\ Z3 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ L/2 \end{vmatrix} \qquad \begin{vmatrix} X4 \\ Z4 \end{vmatrix} = M \begin{vmatrix} W/2 \\ -L/2 \end{vmatrix}$$

(b) Turret coordinates. The turret area is not computed. The turret area is inside the hull area.

(c) Barrier coordinates. The barrier is not computed. The barrier does not cover the target area.

(d) Aimpoint coordinates.

$$\begin{vmatrix} X0 \\ Z0 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

(e) Equations for the sides (See figure 2).

$$L1(X) = Z1 + \tan(A2) * (X - X1)$$
$$L2(X) = Z2 + \tan(A1) * (X - X2)$$
$$R1(X) = Z1 + \tan(A1) * (X - X1)$$
$$R2(X) = Z2 + \tan(A2) * (X - X2)$$

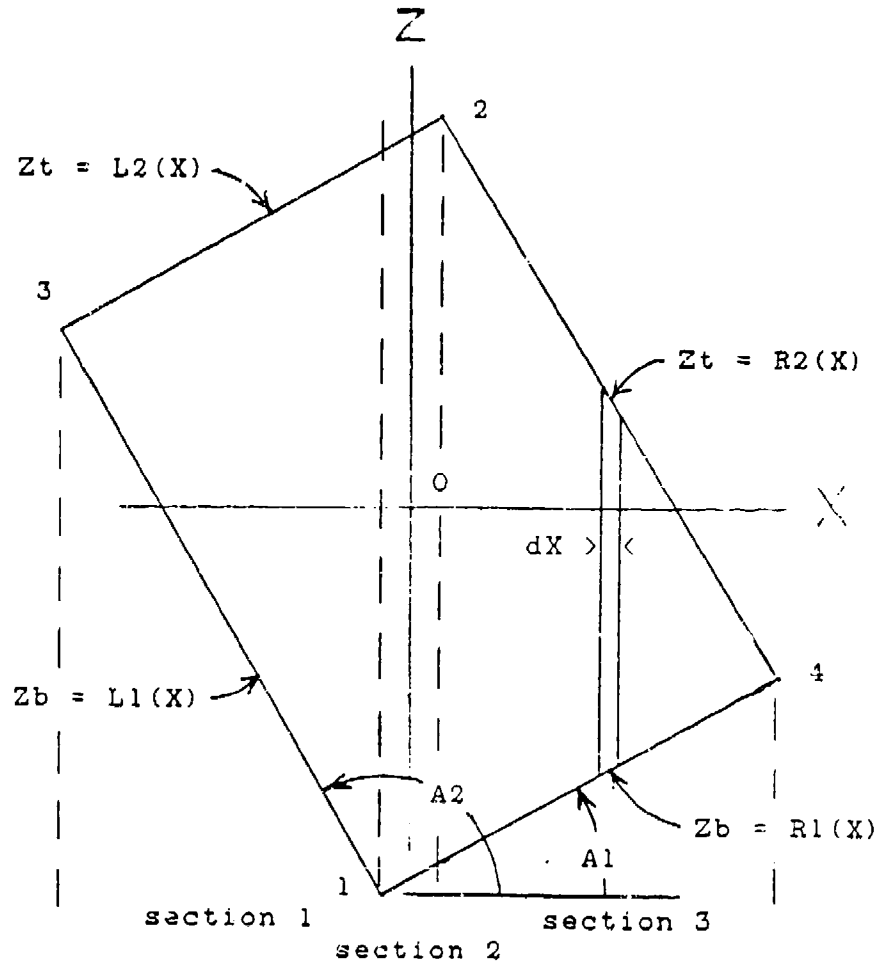

Figure 2. Straight-down area

(2) Ph formula.

If X1 < X2 (Shown in figure 2.),

(Section 1) (Section 2) (Section 3)

$$Ph = \int_{X3}^{X1}\int_{L1(X)}^{L2(X)} p(X,Z)\, dZ\, dX + \int_{X1}^{X2}\int_{R1(X)}^{L2(X)} p(X,Z)\, dZ\, dX + \int_{X2}^{X4}\int_{R1(X)}^{R2(X)} p(X,Z)\, dZ\, dX$$

If X2 < X1,

(Section 1) (Section 2) (Section 3)

$$Ph = \int_{X3}^{X2}\int_{L1(X)}^{L2(X)} p(X,Z)\, dZ\, dX + \int_{X2}^{X1}\int_{L1(X)}^{R2(X)} p(X,Z)\, dZ\, dX + \int_{X1}^{X4}\int_{R1(X)}^{R2(X)} p(X,Z)\, dZ\, dX$$

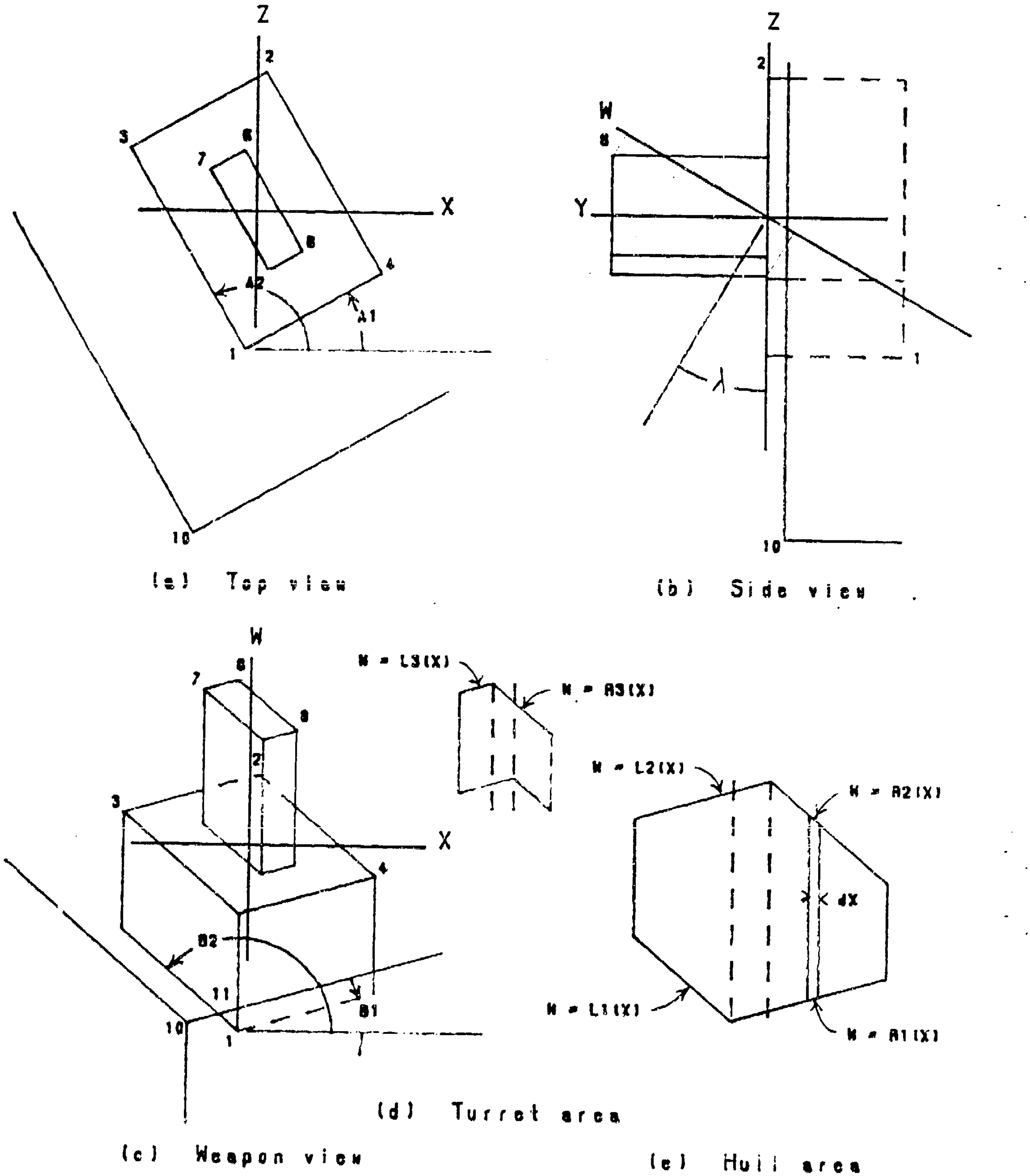

Figure 3. Angle area

d. Angle attack.

(1) Target area formulas.

(a) Coordinates.

1. DFWP first computes the X-Z plane.

a. Let: A1 = aspect angle
A2 = A1 + 90
M = rotation matrix = $\begin{vmatrix} \cos(A1) & -\sin(A1) \\ \sin(A1) & \cos(A1) \end{vmatrix}$

b. Hull corner coordinates.

$$\begin{vmatrix} X1 \\ Z1 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ -L/2 \end{vmatrix} \qquad \begin{vmatrix} X2 \\ Z2 \end{vmatrix} = M \begin{vmatrix} W/2 \\ L/2 \end{vmatrix}$$

$$\begin{vmatrix} X3 \\ Z3 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ L/2 \end{vmatrix} \qquad \begin{vmatrix} X4 \\ Z4 \end{vmatrix} = M \begin{vmatrix} W/2 \\ -L/2 \end{vmatrix}$$

c. Turret corner coordinates.

$$\begin{vmatrix} X6 \\ Z6 \end{vmatrix} = M \begin{vmatrix} w/2 \\ l/2 \end{vmatrix}$$

$$\begin{vmatrix} X7 \\ Z7 \end{vmatrix} = M \begin{vmatrix} -w/2 \\ l/2 \end{vmatrix} \qquad \begin{vmatrix} X8 \\ Z8 \end{vmatrix} = M \begin{vmatrix} w/2 \\ -l/2 \end{vmatrix}$$

d. Barrier corner coordinates.

$$\begin{vmatrix} X10 \\ Z10 \end{vmatrix} = M \begin{vmatrix} -(W/2+H) \\ -(L/2+H) \end{vmatrix}$$

2. DFWP then computes the Y-Z plane.

a. Y coordinates. See paragraph 4b(1).

b. Z coordinates. See paragraph 4d(1)(a)1.

3. DFWP then computes the X-W plane (target area plane).

a. X coordinates. See paragraph 4d(1)(a)1.

b. W coordinates. Figure 3 (b) shows the side view of the target (Y-Z plane). It shows the angle ,λ, of flight path and the W-axis which is the edge of X-W plane perpendicular to the flight path. The Y and Z coordinates and λ compute the W coordinate by:

$$W\% = Z\% * \sin(\lambda) + Y\% * \cos(\lambda).$$

(b) Barrier coverage. The methodology computes the hull area that the barrier does not cover.

1. Ways barrier covers hull. The position of the barrier corner 10 in relation to the bottom edges of the hull determines the way the barrier covers the hull.

- Case 1. Point 10 is below both edges. Neither side of barrier covers hull.

- Case 2. Point 10 is above both edges. Both sides of barrier cover hull.

- Case 3. Point 10 is above right edge and below left edge. Right side of barrier covers hull. (This is the case in figure 3 (c).)

- Case 4. Point 10 is below right edge and above left edge. Left side of barrier covers hull.

2. Determination of way barrier covers hull. Points 1 and 10 get U and V coordinates. The U-axis is perpendicular to the left bottom edge of hull, and the V-axis is perpendicular to the right bottom edge of hull. See figure 4.

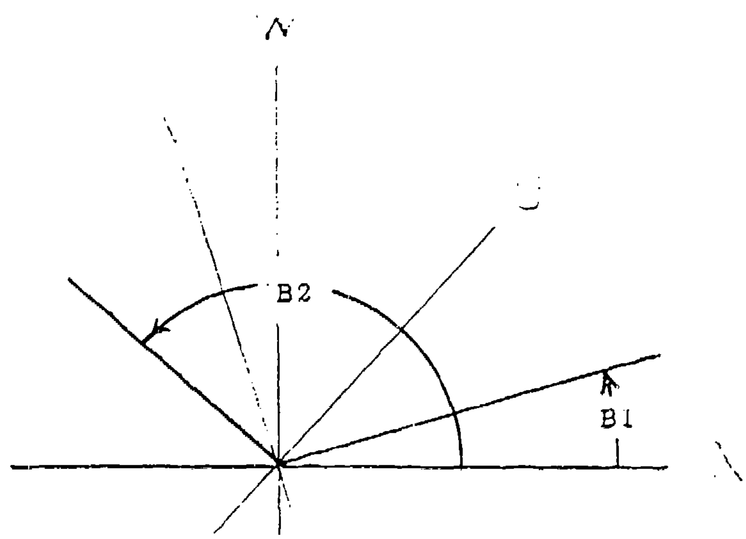

Figure 4. U and V axes

Let U be angle of U axis and V be angle of V axis.

Then:

$$\begin{aligned} \cos(U) &= \cos(B2-90) &&= \sin(B2) \\ \sin(U) &= \sin(B2-90) &&= -\cos(B2) \\ \cos(V) &= \cos(B1+90) &&= -\sin(B1) \\ \sin(V) &= \sin(B1+90) &&= \cos(B1) \end{aligned}$$

Let:

$$M = \begin{vmatrix} \cos(U) & \sin(U) \\ \cos(V) & \sin(V) \end{vmatrix}$$

Then:

$$\begin{vmatrix} U10 \\ V10 \end{vmatrix} = M \begin{vmatrix} X10 \\ W10 \end{vmatrix} \qquad \begin{vmatrix} U1 \\ V1 \end{vmatrix} = M \begin{vmatrix} X1 \\ W1 \end{vmatrix}$$

- Case 1. $U10 < U1$ and $V10 < V1$.

- Case 2. $U10 > U1$ and $V10 > V1$.

- Case 3. $U10 < U1$ and $V10 > V1$.

- Case 4. $U10 > U1$ and $V10 < V1$.

3. Computation. Point 1, the bottom point of hull, is changed to the bottom point of the part of hull that the barrier does not cover.

- Case 1. Point 1 is not changed.

- Case 2. Point 1 is changed to the barrier point 10.

- Case 3. Point 1 is changed to the point 11. Point
11 is the intersection of right top edge of barrier with bottom left edge of hull. Coordinates of point 11 are:

$$\begin{aligned} X11 &= (\tan(B2)*X1 - \tan(B1)*X10 + W10 - W1) \\ &\quad / (\tan(B2) - \tan(B1)) \\ W11 &= (\cot(B2)*W1 - \cot(B1)*W10 + X10 - X1) \\ &\quad / (\cot(B2) - \cot(B1)) \end{aligned}$$

- Case 4. Point 1 is changed to the point 12. Point
12 is the intersection of left top edge of barrier with bottom right edge of hull. Coordinates of point 12 are:

$$\begin{aligned} X12 &= (\tan(B2)*X10 - \tan(B1)*X1 + W1 - W10) \\ &\quad / (\tan(B2) - \tan(B1)) \\ W12 &= (\cot(B2)*W10 - \cot(B1)*W1 + X1 - X10) \\ &\quad / (\cot(B2) - \cot(B1)) \end{aligned}$$

(c) Turret coverage. The methodology computes the turret area that does not cover the hull.

1. Ways turret extends over hull. The position of point 6 in relation to the top edges of the hull determines the way the turret extends over the hull.

- Case 1. Point 6 is below both edges. The turret is within the hull area.

area

- Case 2. Point 6 is above both edges. The turret may extend over both edges.

- Case 3. Point 6 is below right edge and above left edge. The turret extends above left edge.

- Case 4. Point 6 is above right edge and below left edge. The turret extends above right edge.

2. Determination of way turret extends over hull. Point 6 and point 2 get U and V coordinates. Let U, V, and M be the same as in paragraph 4d(1)(b)2.

Then: $\begin{vmatrix} U6 \\ V6 \end{vmatrix} = M \begin{vmatrix} X6 \\ W6 \end{vmatrix} \quad \begin{vmatrix} U2 \\ V2 \end{vmatrix} = M \begin{vmatrix} X2 \\ W2 \end{vmatrix}$

- Case 1. $U6 < U2$ and $V6 < V2$.

- Case 2. $U6 > U2$ and $V6 > V2$.

- Case 3. $U6 < U2$ and $V6 > V2$.

- Case 4. $U6 > U2$ and $V6 < V2$.

3. Computation. The X coordinate of a point 5, which is on the bottom part of turret not covering hull, is computed.

- Case 1. X5 is not computed.

- Case 2. If $X7 < X2 < X8$ (Turret extends over both edges of hull.), then X5 is made equal to X2. If $X8 < X2$ (Turret extends over left edge of hull.), then X5 is made equal to X8. If $X2 < X7$ (Turret extends over right edge of hull.), then X5 is made equal to X7.

- Case 3. X21 is computed. Point 21 is the intersection of top right edge of turret with top left edge of hull.

  X21 = (tan(B2)* X2 - tan(B1)* X6 + W6 - W2) / (tan(B2) - tan(B1))

  If X8 < X21 (Top edge of turret does not intersect with top edge of hull.), then X5 is made equal to X8. Otherwise X5 is made equal to X21.

- Case 4. X22 is computed. Point 22 is the intersection of top left edge of turret with top right edge of hull.

  X22 = (tan(B2)* X6 - tan(B1)* X2 + W2 - W6) / (tan(B2) - tan(B1))

  If X22 < X7 (Top edge of turret does not intersect with top edge of hull.), then X5 is made equal to X7. Otherwise X5 is made equal to X22.

4. Angles. A1 and A2 are labeled B1 and B2 in the X-W plane. See figure 3 (c). Figure 5 shows the cosines and sines of A1 and A2 in the X-Z plane and of B1 and B2 in the X-W plane. Z distances are changed to W distances by the formula

$$W = Z * \sin(\lambda)$$

Therefore the formulas

```
D1     =  sqrt (cos(A1)**2 + (sin(λ)*sin(A1)**2)
cos(B1)  =  cos(A1) / D1
sin(B1)  =  sin(λ) * sin(A1) / D1
tan(B1)  =  sin(B1) / cos(B1)

D2     =  sqrt (cos(A2)**2 + (sin(λ)*sin(A2)**2)
cos(B2)  =  cos(A2) / D2
sin(B2)  =  sin(λ) * sin(A2) / D2
tan(B2)  =  sin(B2) / cos(B2)
```

compute B1 and B2.

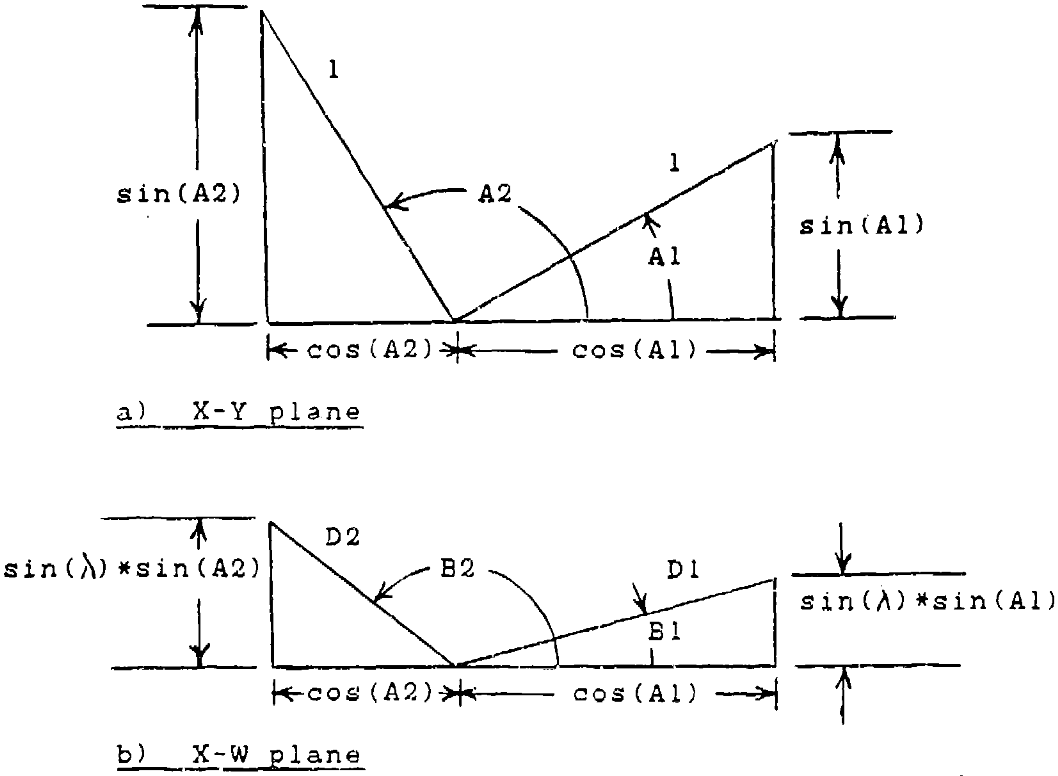

Figure 5. Angles of target sides

5. Equations of the sides. (See figure 3.)

L1(X) = W1 + tan(B2) * (X - X1)
L2(X) = W2 + tan(B1) * (X - X2)
L3(X) = W6 + tan(B1) * (X - X6)

R1(X) = W1 + tan(B1) * (X - X1)
R2(X) = W2 + tan(B2) * (X - X2)
R3(X) = W6 + tan(B2) * (X - X6)

(2) Probability of hit formulas.

(a) Ph on hull.

If $X1 < X2$ (This is the case in figure 3.).

(Section 1) (Section 2)

$$Ph = \int_{X3}^{X1} f(X) \int_{L1(X)}^{L2(X)} f(Z)\, dZ\, dX + \int_{X1}^{X2} f(X) \int_{R1(X)}^{L2(X)} f(Z)\, dZ\, dX$$

(Section 3)

$$+ \int_{X2}^{X4} f(X) \int_{R1(X)}^{R2(X)} f(Z)\, dZ\, dX$$

If $X2 < X1$.

$$Ph = \int_{X3}^{X2} f(X) \int_{L1(X)}^{L2(X)} f(Z)\, dZ\, dX + \int_{X2}^{X1} f(X) \int_{L1(X)}^{R2(X)} f(Z)\, dZ\, dX$$

$$+ \int_{X1}^{X4} f(X) \int_{R1(X)}^{R2(X)} f(Z)\, dZ\, dX$$

(b) Ph on turret.

If $X5 < X6$.

(Section 1) (Section 2)

$$Ph = \int_{X7}^{X5} f(X) \int_{L2(X)}^{L3(X)} f(Z)\, dZ\, dX + \int_{X5}^{X6} f(X) \int_{R2(X)}^{L3(X)} f(Z)\, dZ\, dX$$

(Section 3)

$$+ \int_{X6}^{X8} f(X) \int_{R2(X)}^{R3(X)} f(Z)\, dZ\, dX$$

If X6 < X5 (This is the case in figure 3.),

$$Ph = \int_{X7}^{X6} f(X) \int_{L2(X)}^{L3(X)} f(Z)\, dZ\, dX + \int_{X6}^{X5} f(X) \int_{L2(X)}^{R3(X)} f(Z)\, dZ\, dX$$

$$+ \int_{X5}^{X8} f(X) \int_{R2(X)}^{R3(X)} f(Z)\, dZ\, dX$$

(c) Ph on target.

Ph = Ph on hull + Ph on turret

5. <u>Probability of hit</u>.

a. Level attack. Each double integral in the Ph formula from paragraph 4b(2) is able to be changed into a product of two single integrals, thus:

$$Ph = \int_{X3}^{X4} f(X)\, dX * \int_{Y10}^{0} f(Y)\, dY + \int_{X7}^{X8} f(X)\, dX * \int_{0}^{h} f(Y)\, dY$$

Each integral can be evaluated from two lookups in the normal distribution table.

b. Angle or straight-down attack. From paragraphs 4c(2) and 4d(2), the Ph formula for a section is:

$$Ph = \int_{X\%}^{X\%} f(X) \int_{K1(X)}^{K2(X)} f(Y)\, dY\, dX$$

The outer integral must be evaluated numerically. The inner integral, though, for an X value can be evaluated from two lookups in the normal distribution table. The numerical integration method that DFWP implements is: First DFWP makes a change of variable, dU = f(X) dX. This gets rid of f(X) in the integrand. The integral becomes:

$$Ph = \int_{U\%}^{U\%} \int_{K1(X)}^{K2(X)} f(Y)\, dY\, dU$$

The normal distribution table gives the outer integral limits.

$$U\% = \int_{-\infty}^{X\%} f(T)\, dT = F(X\%)$$

The inverse normal table gives the Xs of the inner integral limits.

$$X = F^{-1}(U)$$

The DWFP then does the numerical evaluation using Us with values .005, .015, .025, etc., .995, which are between the outer integral limits. These values are midpoints of intervals of width .01 which are [0.,.01], [.01,.02], [.02,.03], etc., [.99,1.]. The beginning interval and last interval of the numerical integration will not necessarily have widths of .01.

6. <u>Example computation</u>.

a. Input values.

W = Hull width = 2 m	w = Turret width = .5 m
L = Hull length = 3 m	l = Turret length = 1.5 m
H = Hull height = 1.75 m	h = Turret height = 2 m
Hb = Barrier height = 1.5 m	
A1 = Aspect angle = 30	A2 = A1 + 90 = 120
XB = X-bias = 0.2 m	WB = W-bias = 0.
XD = X-dispersion = 2 m	WD = W-dispersion = 1 m

b. Level attack.

(1) Target area.

(a) Hull.

$$\begin{aligned} X4 &= W/2 * \cos(A1) + L/2 * \sin(A1) \\ &= 1. * \cos(30) + 1.5 * \sin(30) \\ &= 1.616 \text{ m} \\ X3 &= -X4 \\ &= -1.616 \text{ m} \end{aligned}$$

(b) Turret.

$$X8 = w/2 * \cos(A1) + l/2 * \sin(A1)$$
$$= 0.25 * \cos(30) + 0.75 * \sin(30)$$
$$= 0.5915 \text{ m}$$
$$X7 = -X8$$
$$= -0.5915 \text{ m}$$

(c) Barrier.

$$Y10 = -H + Hb = -0.25 \text{ m}$$

(d) Aimpoint.

$$X0 = 0.$$
$$Y0 = (h + Y10) / 2$$
$$= 0.875 \text{ m}$$

(2) Ph Integral.

$$Ph = \int_{X3}^{X4} f(X)\,dX * \int_{Y10}^{0} f(Y)\,dY + \int_{X7}^{X8} f(X)\,dX * \int_{0}^{h} f(Y)\,dY$$

The integral limits must be changed to standard deviations by the formulas: X-deviation = (X - XM) / XD and the same for Y.

The mean is the sum of aimpoint and bias.

$$XM = X0 + XB = 0. + 0.2 = 0.2 \text{ m}$$
$$YM = Y0 + YB = .875 + 0. = 0.875 \text{ m}$$

First upper limit = (X4 -0.2) / 2. = 0.708 sd
First lower limit = (X3 -0.2) / 2. = -0.908 sd

Second upper limit = (0 -0.875) / 1. = -0.875 sd
Second lower limit = (Y10 -0.875) /1. = -1.125 sd

Third upper limit = (X8 -0.2) / 2. = 0.19575 sd
Third lower limit = (X7 -0.2) / 2. = -0.39575 sd

Fourth upper limit = (h -0.875) / 1. = 1.125 sd
Fourth lower limit = (0 -0.875) / 1. = -0.875 sd

Using the normal distribution table F(X).

	Ph left of hull right side		Ph left of hull left side		Ph below hull top side		Ph below hull bottom side
Ph = (	F(0.708)	-	F(-0.908))	*	(F(-0.875)	-	F(-1.125))

$$+ \underbrace{(F(0.19575)}_{\text{Ph left of turret right side}} - \underbrace{F(-0.39575))}_{\text{Ph left of turret left side}} * \underbrace{(F(1.125)}_{\text{Ph below turret top side}} - \underbrace{F(-0.875))}_{\text{Ph below turret bottom side}}$$

$$Ph = \underbrace{(.76053 - .18194)}_{\text{Ph between hull right and left sides}} * \underbrace{(.19079 - .13029)}_{\text{Ph between hull top and bottom sides}}$$

$$+ \underbrace{(.57760 - .34614)}_{\text{Ph between turret right and left sides}} * \underbrace{(.86971 - .19079)}_{\text{Ph between turret top and bottom sides}}$$

$$Ph = .57859 * .06050 + .23146 * .67892$$
$$= .19215$$

c. Straight-down attack.

(1) Target area.

(a) Coordinates.

1. In meters.

$$M = \begin{vmatrix} \cos(30) & -\sin(30) \\ \sin(30) & \cos(30) \end{vmatrix}$$

$$\begin{vmatrix} X1 \\ Z1 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ -L/2 \end{vmatrix} = \begin{vmatrix} \cos30*(-1.) - \sin30*(-1.5) \\ \sin30*(-1.) + \cos30*(-1.5) \end{vmatrix}$$

$$= \begin{vmatrix} -0.1160 \text{ m} \\ -1.7990 \text{ m} \end{vmatrix}$$

$$\begin{vmatrix} X2 \\ Z2 \end{vmatrix} = M \begin{vmatrix} W/2 \\ L/2 \end{vmatrix} = \begin{vmatrix} \cos30 * 1. - \sin30 * 1.5 \\ \sin30 * 1. + \cos30 * 1.5 \end{vmatrix}$$

$$= \begin{vmatrix} 0.1160 \text{ m} \\ 1.7990 \text{ m} \end{vmatrix}$$

$$\begin{vmatrix} X3 \\ Z3 \end{vmatrix} = M \begin{vmatrix} -W/2 \\ L/2 \end{vmatrix} = \begin{vmatrix} \cos30*(-1.) - \sin30 * 1.5 \\ \sin30*(-1.) + \cos30 * 1.5 \end{vmatrix}$$

$$= \begin{vmatrix} -1.6160 \text{ m} \\ 0.7990 \text{ m} \end{vmatrix}$$

$$\begin{vmatrix} X4 \\ Z4 \end{vmatrix} = M \begin{vmatrix} W/2 \\ -L/2 \end{vmatrix} = \begin{vmatrix} \cos30 * 1. - \sin30*(-1.5) \\ \sin30 * 1. + \cos30*(-1.5) \end{vmatrix}$$

$$= \begin{vmatrix} 1.6160 \text{ m} \\ -0.7990 \text{ m} \end{vmatrix}$$

2. In standard deviations.

$$\begin{vmatrix} X1 \\ Z1 \end{vmatrix} = \begin{vmatrix} (-0.116 - XM) / XD \\ (-1.799 - ZM) / ZD \end{vmatrix} = \begin{vmatrix} (-0.116 - 0.2) / 2. \\ (-1.799 - 0.) / 1. \end{vmatrix}$$

$$= \begin{vmatrix} -0.158 \text{ sd} \\ -1.799 \text{ sd} \end{vmatrix}$$

$$\begin{vmatrix} X2 \\ Z2 \end{vmatrix} = \begin{vmatrix} (0.116 - XM) / XD \\ (1.799 - ZM) / ZD \end{vmatrix} = \begin{vmatrix} -0.042 \text{ sd} \\ 1.799 \text{ sd} \end{vmatrix}$$

$$\begin{vmatrix} X3 \\ Z3 \end{vmatrix} = \begin{vmatrix} (-1.616 - XM) / XD \\ (0.799 - ZM) / ZD \end{vmatrix} = \begin{vmatrix} -0.908 \text{ sd} \\ 0.799 \text{ sd} \end{vmatrix}$$

$$\begin{vmatrix} X4 \\ Z4 \end{vmatrix} = \begin{vmatrix} (1.616 - XM) / XD \\ (-0.799 - XM) / ZD \end{vmatrix} = \begin{vmatrix} 0.708 \text{ sd} \\ -0.799 \text{ sd} \end{vmatrix}$$

(b) Equations of sides.

1. In meters.

$$\begin{aligned} L1(X) &= Z1 + \tan(A2) * (X - X1) \\ L2(X) &= Z2 + \tan(A1) * (X - X2) \\ R1(X) &= Z1 + \tan(A1) * (X - X1) \\ R2(X) &= Z2 + \tan(A2) * (X - X2) \end{aligned}$$

2. In standard deviations. The cross section area is different when the X and Z are measured in standard deviations. The angles A1 and A2 are different, and, therefore, tan(A1) and tan(A2) are different. It can be shown that tan(A*) = XD / ZD * tan(A*).

$$\begin{aligned} \tan(A1) &= 2. / 1. * \tan 30 = 1.15470 \\ \tan(A2) &= 2. / 1. * \tan 120 = -3.46410 \\ L1(X) &= -1.799 - 3.4641 * (X + 0.158) \\ L2(X) &= 1.799 + 1.1547 * (X + 0.042) \\ R1(X) &= -1.799 + 1.1547 * (X + 0.158) \\ R2(X) &= 1.799 - 3.4641 * (X + 0.042) \end{aligned}$$

(2) Ph integral.

$$Ph = \int_{X3}^{X1} f(X) \int_{L1(X)}^{L2(X)} f(Z)\, dZ\, dX + \int_{X1}^{X2} f(X) \int_{R1(X)}^{L2(X)} f(Z)\, dZ\, dX$$

$$+ \int_{X2}^{X4} f(X) \int_{R1(X)}^{R2(X)} f(Z)\, dZ\, dX$$

The middle double integral will be evaluated. First a substitution U= F(X), dU = f(X) dX, is made. This changes the integral to:

$$\int_{U1}^{U2} \int_{R1(X)}^{L2(X)} f(Z)\, dZ \quad dU$$

and changes the outer integral limits to:

U1 = F(X1) = F(-0.158) = .43723
U2 = F(X2) = F(-0.042) = .48325

U interval	U	X = F-1(U)	Zt = L2(X)	Zb = R1(X)	Ph below top F(Zt)	Ph below bottom F(Zb)
[.43723,.44]	.435	-0.16366	1.65852	-1.80554	.95139	.03550
[.44, .45]	.445	-0.13830	1.68780	-1.77625	.95428	.03785
[.45, .46]	.455	-0.11304	1.71697	-1.74708	.95701	.04031
[.46, .47]	.465	-0.08785	1.74606	-1.71700	.95960	.04299
[.47, .48]	.475	-0.06271	1.77508	-1.68897	.96206	.04561
[.48,.48325]	.485	-0.03761	1.80407	-1.65999	.96439	.04846

U interval	Ph between top and bottom F(Zt) -F(Zb)	Ph in U interval	Ph in rectangular interval
[.43723,.44]	.9159	.00277	.002537
[.44, .45]	.9164	.01	.009164
[.45, .46]	.9167	.01	.009167
[.46, .47]	.9166	.01	.009166
[.47, .48]	.9164	.01	.009164
[.48,.48325]	.9159	.00325	.002977

Ph = .04218

The Ph for the complete target is given in paragraph 6b.

d. Angle attack.

(1) Target area.

(a) X-Z plane.

1. Hull. Paragraph 6c(1)(a)1 has the computed hull coordinates.

2. Turret.

$$\begin{vmatrix} X6 \\ Z6 \end{vmatrix} = M \begin{vmatrix} w/2 \\ l/2 \end{vmatrix} = \begin{vmatrix} \cos30*.25-\sin30*.75 \\ \sin30*.25+\cos30*.75 \end{vmatrix}$$

$$= \begin{vmatrix} -0.1585 \text{ m} \\ 0.7745 \text{ m} \end{vmatrix}$$

$$\begin{vmatrix} X7 \\ Z7 \end{vmatrix} = M \begin{vmatrix} -w/2 \\ l/2 \end{vmatrix} = \begin{vmatrix} \cos30*(-.25)-\sin30 * .75 \\ \sin30*(-.25)+\cos30 * .75 \end{vmatrix}$$

$$= \begin{vmatrix} -0.5915 \text{ m} \\ 0.5245 \text{ m} \end{vmatrix}$$

$$\begin{vmatrix} X8 \\ Z8 \end{vmatrix} = M \begin{vmatrix} w/2 \\ -l/2 \end{vmatrix} = \begin{vmatrix} \cos30*.25-\sin30*(-.75) \\ \sin30*.25+\cos30*(-.75) \end{vmatrix}$$

$$= \begin{vmatrix} 0.5915 \text{ m} \\ -0.5245 \text{ m} \end{vmatrix}$$

3. Barrier.

$$\begin{vmatrix} X10 \\ Z10 \end{vmatrix} = M \begin{vmatrix} -(W/2+H) \\ -(L/2+H) \end{vmatrix} = \begin{vmatrix} \cos30*(-2.75)-\sin30*(-3.25) \\ \sin30*(-2.75)+\cos30*(-3.25) \end{vmatrix}$$

$$= \begin{vmatrix} -0.7567 \text{ m} \\ -4.1896 \text{ m} \end{vmatrix}$$

(b) Y-Z plane.

1. Y coordinates.

$$\begin{aligned} Y1 &= -H \\ &= -1.75 \text{ m} \\ Y2 &= 0. \\ Y6 &= h \\ &= 2 \text{ m} \\ Y10 &= -H + Hb = -1.75 + 1.5 \\ &= -.25 \text{ m} \\ Y0 &= (h + Y10)/2 = (2. -.25)/2. \\ &= 0.875 \text{ m} \end{aligned}$$

2. Z coordinates. See paragraph 6d(1)(a).

(c) X-W plane.

1. X coordinates. See paragraph 6d(1)(a).

2. W coordinates. $W\% = Z\% * \sin(\ ) + Y\% * \cos(\ )$

$$
\begin{aligned}
W1 &= -1.799 * \sin 30 - 1.75 * \cos 30 \\
&= -2.4150 \text{ m} \\
W2 &= 1.799 * \sin 30 + 0. * \cos 30 \\
&= 0.8995 \text{ m} \\
W6 &= 0.7745 * \sin 30 + 2. * \cos 30 \\
&= 2.1193 \text{ m} \\
W10 &= -4.1896 * \sin 30 - .25 * \cos 30 \\
&= -2.3113 \text{ m} \\
W0 &= 0. * \sin 30 + 0.875 * \cos 30 \\
&= 0.7578 \text{ m}
\end{aligned}
$$

3. Angles.

$$
\begin{aligned}
D1 &= \text{sqrt}\ (\cos(A1)**2 + (\sin(\ ) * \sin(A1))**2) \\
&= \text{sqrt}\ (\cos 30 **2 + (\sin 30 * \sin 30)**2) \\
&= 0.90139 \\
\cos(B1) &= \cos(A1)/D1 = \cos 30 / 0.90139 \\
&= .9608 \\
\sin(B1) &= \sin(\ ) * \sin(A1)/D1 \\
&= \sin 30 * \sin 30 / 0.90139 \\
&= 0.27735 \\
\tan(B1) &= \sin(B1) / \cos(B1) \\
&= 0.2887
\end{aligned}
$$

$$
\begin{aligned}
D2 &= \text{sqrt}\ (\cos(A2)**2 + (\sin(\ ) * \sin(A2)**2) \\
&= \text{sqrt}\ (\cos 120 **2 + (\sin 30 * \sin 120)**2) \\
&= 0.66144 \\
\cos(B2) &= \cos(A2)/D2 = \cos 120 / 0.66144 \\
&= -0.7559 \\
\sin(B2) &= \sin(\ ) * \sin(A2)/D2 \\
&= \sin 30 * \sin 120 / 0.66144 \\
&= 0.65465 \\
\tan(B2) &= \sin(B2) / \cos(B2) \\
&= -0.86605
\end{aligned}
$$

4. Barrier coverage.

a. U-V coordinates.

$$
M = \begin{vmatrix} \cos(U) & \sin(U) \\ \cos(V) & \sin(V) \end{vmatrix} = \begin{vmatrix} \sin(B2) & -\cos(B2) \\ -\sin(B1) & \cos(B1) \end{vmatrix}
$$

$$
= \begin{vmatrix} 0.65465 & 0.7559 \\ -0.27735 & 0.9608 \end{vmatrix}
$$

$$
\begin{vmatrix} U10 \\ V10 \end{vmatrix} = M \begin{vmatrix} X10 \\ W10 \end{vmatrix} = \begin{vmatrix} .65465*(-.7567)+.7559*(-2.3113) \\ -.27735*(-.7567)+.9608*(-2.3113) \end{vmatrix}
$$

$$
= \begin{vmatrix} -2.2425 \text{ m} \\ -2.0108 \text{ m} \end{vmatrix}
$$

$$\begin{vmatrix} U1 \\ V1 \end{vmatrix} = M \begin{vmatrix} X1 \\ W1 \end{vmatrix} = \begin{vmatrix} .65465*(-0.116)+.7559*(-2.415) \\ -.27735*(-0.116)+.9608*(-2.415) \end{vmatrix}$$

$$= \begin{vmatrix} -1.9014 \text{ m} \\ -2.2882 \text{ m} \end{vmatrix}$$

b. Coverage. U10 < U1 and V10 > V1. Case equals 3. Right side of barrier covers hull and intersects with left side of hull at point 11.

$$\begin{aligned} X11 &= (\tan(B2) * X1 - \tan(B1) * X10 + W10 - W1) \\ &\quad / (\tan(B2) - \tan(B1)) \\ &= (-0.866 *(-0.116) - 0.2887 *(-0.7567) - \\ &\quad 2.3113 + 2.415) / (-0.866 - 0.2887) \\ &= -0.3660 \text{ m} \end{aligned}$$

$$\begin{aligned} W11 &= (W1 / \tan(B2) - W10 / \tan(B1) + X10 - X1) \\ &\quad / (1 / \tan(B2) - 1 / \tan(B1)) \\ &= (-2.415 /(-0.866) + 2.3113 /0.2887 - 0.7567 + \\ &\quad 0.116) / (1 /(-0.866) - 1 /0.2887) \\ &= -2.1985 \text{ m} \end{aligned}$$

5. Turret coverage.

a. U-V coordinates.

$$\begin{vmatrix} U6 \\ V6 \end{vmatrix} = M \begin{vmatrix} X6 \\ W6 \end{vmatrix} = \begin{vmatrix} .65465*(-0.1585)+.7559*2.1193 \\ -.27735*(-0.1585)+.9608*2.1193 \end{vmatrix}$$

$$= \begin{vmatrix} 1.4982\text{m} \\ 2.0802\text{m} \end{vmatrix}$$

$$\begin{vmatrix} U2 \\ V2 \end{vmatrix} = M \begin{vmatrix} X2 \\ W2 \end{vmatrix} = \begin{vmatrix} .65465*0.116+.7559*0.8995 \\ -.27735*0.116+.9608*0.8995 \end{vmatrix}$$

$$= \begin{vmatrix} 0.7559\text{m} \\ 0.8321\text{m} \end{vmatrix}$$

b. Extention over hull. U6 > U2 and V6 > V2. Case equals 2. The turret area may extend above both left and right edges of hull area. X7 < X6 < X8; therefore, the turret area does extend above both edges of hull area. Make bottom point 5 of turret area the top point 2 of hull.

(2) Ph integral. The Ph is given in paragraph 7b.

7. <u>Computer program Top_ph</u>. Computer program 'Top_Ph' uses the new methodology in DFWP to compute Ph for sample weapons against sample targets. Inputs are the target dimensions, aspect angle, attack angle, and probability distribution data for the weapon.

Program Top_Ph uses two methods for the integration, one being the method in DFWP. For level attack, the methods are table lookup, the method in DFWP, and the numerical method. The numerical method in DFWP is done by subroutine SUMMATE. For angle attack, the integration methods are done by subroutines SUMMATE and TRAPEZOID. TRAPEZOID does the integration by using the trapezoid method and without the change of variable. TRAPEZOID is made very accurate by doing 10,000 intervals.

APPENDIX A

TOP_PH OUTPUT

```
HULL WIDTH, LENGTH, and HEIGHT?
  2.00   3.00   1.75
TURRET WIDTH, LENGTH, and HEIGHT?
  0.50   1.50   2.00
BARRIER HEIGHT?
  1.50
ASPECT ANGLE and ATTACK ANGLE?
 30.    0.
XBIAS, XDISPERSION?
  0.20   2.00
WBIAS, WDISP?
  0.00   1.00
                LOOKUP     SUMMATE
PH HULL    =    0.03500    0.3500
PH TURRET  =    0.15714    0.15714
PH         =    0.19214    0.19214
```

```
HULL WIDTH, LENGTH, and HEIGHT?
  2.00   3.00   1.75
TURRET WIDTH, LENGTH, and HEIGHT?
  0.50   1.50   2.00
BARRIER HEIGHT?
  1.50
ASPECT ANGLE and ATTACK ANGLE?
 30.  90.
XBIAS, XDISPERSION?
  0.20   2.00
WBIAS, WDISP?
  0.00   1.00
        SUMMATE   TRAPEZOID
PH1 =   0.14498   0.14499
PH2 =   0.04219   0.04219
PH3 =   0.15423   0.15422
PH  =   0.34140   0.34140

HULL WIDTH, LENGTH, and HEIGHT?
  2.00   3.00   1.75
TURRET WIDTH, LENGTH, and HEIGHT?
  0.50   1.50   2.00
BARRIER HEIGHT?
  1.50
ASPECT ANGLE and ATTACK ANGLE?
 30.  30.
XBIAS, XDISPERSION?
  0.20   2.00
WBIAS, WDISP?
  0.00   1.00
        SUMMATE   TRAPEZOID
  HULL
PH1 =   0.11273   0.11273
PH2 =   0.02490   0.02490
PH3 =   0.08920   0.08919
PH  =   0.22683   0.22682
  TURRET
PH1 =   0.03332   0.03332
PH2 =   0.01912   0.01913
PH3 =   0.03248   0.03248
PH  =   0.08492   0.08493
  TARGET
PH  =   0.31175   0.31174
```

APPENDIX B

TOP_PH CODE

```
      PROGRAM  PH
!
      REAL  Lh, Lt
      DATA  DTR /.01745329/
      COMMON /TA/  XBOT, WBOT, XTOP, WTOP, XMEAN, XDISP, WMEAN,
      1   WDISP
      COMMON /ROT/  COS_A1, SIN_A1
!
C ** INPUT
      WRITE (*) ' HULL WIDTH, LENGTH, and HEIGHT?  '
      WRITE (1,'(A)') ' HULL WIDTH, LENGTH, and HEIGHT?  '
      READ (*,'(3F7.0)') Wh, Lh, Hh
      WRITE (1,'(3F7.2)') Wh, Lh, Hh
      WRITE (*) ' TURRET WIDTH, LENGTH, and HEIGHT?  '
      WRITE (1,'(A)') ' TURRET WIDTH, LENGTH, and HEIGHT?  '
      READ (*,'(3F7.0)') Wt, Lt, Ht
      WRITE (1,'(3F7.2)') Wt, Lt, Ht
      WRITE (*) ' BARRIER HEIGHT?  '
      WRITE (1,'(A)') ' BARRIER HEIGHT?  '
      READ (*,'(F7.0)') Hb
      WRITE (1,'(F7.2)') Hb
      WRITE (*) ' ASPECT ANGLE and ATTACK ANGLE?  '
      WRITE (1,'(A)') ' ASPECT ANGLE and ATTACK ANGLE?  '
      READ (*,'(2F5.0)') A1, TA
      WRITE (1,'(2F5.0)') A1, TA
      WRITE (*) ' XBIAS, XDISPERSION?  '
      WRITE (1,'(A)') ' XBIAS, XDISPERSION?  '
      READ (*,'(2F7.0)') XBIAS, XDISP
      WRITE (1,'(2F7.2)') XBIAS, XDISP
      WRITE (*) ' WBIAS, WDISP?  '
      WRITE (1,'(A)') ' WBIAS, WDISP?  '
      READ (*,'(2F7.0)') WBIAS, WDISP
      WRITE (1,'(2F7.2)') WBIAS, WDISP
!
C ** ANGLES
      COS_A1 = COS (DTR * A1)
      SIN_A1 = SIN (DTR * A1)
      TAN_A1 = TAN (DTR * A1)
!
      COS_A2 = -SIN_A1
      SIN_A2 = COS_A1
      TAN_A2 = SIN_A2 / COS_A2
!
      COS_TA = COS (DTR * TA)
      SIN_TA = SIN (DTR * TA)
!
C ** COORDINATES
C **  AIMPOINT
```

```
        X0 = 0.
        Y0 = (Ht - Hh + Hb) / 2.
        Z0 = 0.
C ** HULL
        CALL  ROTATE (-Wh/2., -Lh/2., X1, Z1)
        Y1 = -Hh
        CALL  ROTATE (Wh/2., Lh/2., X2, Z2)
        Y2 = 0.
        CALL  ROTATE (-Wh/2., Lh/2., X3, Z3)
        CALL  ROTATE (Wh/2., -Lh/2., X4, Z3)
C ** TURRET
        CALL  ROTATE (Wt/2., Lt/2., X6, Z6)
        Y6 = Ht
        CALL  ROTATE (-Wt/2., Lt/2., X7, Z7)
        CALL  ROTATE (Wt/2., -Lt/2., X8, Z8)
C ** BARRIER
        CALL  ROTATE (-Wh/2.-Hh, -Lh/2.-Hh, X10, Z10)
        Y10 = -Hh + Hb
!
C **********  LEVEL ATTACK  **********
        IF (TA .EQ. 0.) THEN
            XMEAN = X0 + XBIAS
            WMEAN = Y0 + WBIAS
C     ** HULL
            PH_HULL =
     1      (DFN ((X4-XMEAN) /XDISP) - DFN ((X3-XMEAN) /XDISP)) *
     2      (DFN ((0.-WMEAN) /WDISP) - DFN ((Y10-WMEAN) /WDISP))

            WBOT = Y10
            WTOP = 0.
            CALL  SUMMATE (X3, X4, 0., 0., PH_HULL_SUM)
C     ** TURRET
            PH_TURR =
     1      (DFN ((X8-XMEAN) /XDISP) - DFN ((X7-XMEAN) /XDISP)) *
     2      (DFN ((Y6-WMEAN) /WDISP) - DFN ((0.-WMEAN) /WDISP))
            WBOT = 0.
            WTOP = Y6
            CALL  SUMMATE (X7, X8, 0., 0., PH_TURR_SUM)
            WRITE (1,'(A/3(A,2F9.4/))')
     1      '                LOOKUP   SUMMATE',
     2      ' PH HULL   =', PH_HULL, PH_HULL_SUM,
     3      ' PH TURRET =', PH_TURR, PH_TURR_SUM,
     4      ' PH        =', PH_HULL+PH_TURR,
     5      PH_HULL_SUM+PH_TURR_SUM
!
C **********  STRAIGHT DOWN ATTACK  **********
        ELSE IF (TA .EQ. 90.) THEN
            XMEAN = X0 + XBIAS
            WMEAN = Z0 + WBIAS
            XBOT = X1
            WBOT = Z1
            XTOP = X2
```

```
            WTOP = Z2
C    ** DIVIDE HULL into THREE SECTIONS
            IF (X1 .LT. X2) THEN
                XL = X1
                XR = X2
                TAN_A_ = TAN_A1
            ELSE
                XL = X2
                XR = X1
                TAN_A_ = TAN_A2
            ENDIF
C     ** PH on LEFT SECTION
            CALL  SUMMATE (X3, XL, TAN_A2, TAN_A1, PH1_SUM)
            CALL  TRAPEZOID (X3, XL, TAN_A2, TAN_A1, PH1_TRAP)
C     ** PH on MIDDLE SECTION
            CALL  SUMMATE (XL, XR, TAN_A_, TAN_A_, PH2_SUM)
            CALL  TRAPEZOID (XL, XR, TAN_A_, TAN_A_, PH2_TRAP)
C     ** PH on RIGHT SECTION
            CALL  SUMMATE (XR, X4, TAN_A1, TAN_A2, PH3_SUM)
            CALL  TRAPEZOID (XR, X4, TAN_A1, TAN_A2, PH3_TRAP)
            WRITE (1,'(A/4(A,2F10.5/))')
        1   '            SUMMATE   TRAPEZOID',
        2   ' PH1 =', PH1_SUM, PH1_TRAP,
        3   ' PH2 =', PH2_SUM, PH2_TRAP,
        4   ' PH3 =', PH3_SUM, PH3_TRAP,
        5   ' PH  =', PH1_SUM + PH2_SUM + PH3_SUM,
        6               PH1_TRAP + PH2_TRAP + PH3_TRAP
!
C **********  ANGLE ATTACK  **********
        ELSE
C     **  W COORDINATES
            W0 = Y0 * COS_TA + Z0 * SIN_TA
            W1 = Y1 * COS_TA + Z1 * SIN_TA
            W2 = Y2 * COS_TA + Z2 * SIN_TA
            W6 = Y6 * COS_TA + Z6 * SIN_TA
            W10 = Y10 * COS_TA + Z10 * SIN_TA
C     **  ANGLES
            D1 = SQRT (COS_A1 **2 + (SIN_TA * SIN_A1) **2)
            COS_B1 = COS_A1 / D1
            SIN_B1 = SIN_TA * SIN_A1 / D1
            TAN_B1 = SIN_B1 / COS_B1
            COS_V = -SIN_B1
            SIN_V = COS_B1
!
            D2 = SQRT (COS_A2 **2 + (SIN_TA * SIN_A2) **2)
            COS_B2 = COS_A2 / D2
            SIN_B2 = SIN_TA * SIN_A2 / D2
            TAN_B2 = SIN_B2 / COS_B2
            COS_U = SIN_B2
            SIN_U = -COS_B2
C     ** MEANS
            XMEAN = X0 + XBIAS
```

```
      WMEAN = W0 + WBIAS
!
C     ***** HULL *****
C     ** POINT 1 and BARRIER POINT in U-V COORDINATES.
      U1 = X1 * COS_U + W1 * SIN_U
      V1 = X1 * COS_V + W1 * SIN_V
!
      U10 = X10 * COS_U + W10 * SIN_U
      V10 = X10 * COS_V + W10 * SIN_V
C     **  DETERMINE SITUATION of BARRIER COVERAGE to DETERMINE
C     **  BOTTOM POINT of EXPOSED HULL.
      IF (U10 .GT. U1 .AND. V10 .GT. V1) THEN
C       ** BARRIER PROJECTS OVER BOTH BOTTOM EDGES OF HULL.
          XBOT = X10
          WBOT = W10
      ELSE IF (V10 .GT. V1) THEN
C       ** BARRIER PROJECTS over RIGHT EDGE of HULL.
C       ** COMPUTE INTERSECTION of LEFT EDGE of HULL with RIGHT
C       ** EDGE of BARRIER.
          XBOT = (TAN_B2 * X1 - TAN_B1 * X10 + W10 - W1)
     1              / (TAN_B2 - TAN_B1)
          WBOT = (W1 / TAN_B2 - W10 / TAN_B1 + X10 - X1)
     1              / (1./TAN_B2 - 1./TAN_B1)
        ELSE IF (U10 .GT. U1) THEN
C       ** BARRIER PROJECTS over LEFT EDGE of HULL.  COMPUTE
C       ** INTERSECTION of RIGHT EDGE of HULL with LEFT EDGE
C       ** of BARRIER.
          XBOT = (TAN_B1 * X1 - TAN_B2 * X10 + W10 - W1)
     1              / (TAN_B1 - TAN_B2)
          WBOT = (W1 / TAN_B1 - W10 / TAN_B2 + W10 - X1)
     1                 / (1./TAN_B1 - 1./TAN_B2)
      ELSE
C       ** BARRIER PROJECTS under HULL.
          XBOT = X1
          WBOT = W1
      ENDIF
      XTOP = X2
      WTOP = W2
!
C     ** DIVIDE EXPOSED HULL into THREE SECTIONS
      IF (X1 .LT. X2) THEN
          XL = X1
          XR = X2
          TAN_B_ = TAN_B1
      ELSE
          XL = X2
          XR = X1
          TAN_B_ = TAN_B2
      ENDIF
C     ** PH on LEFT SECTION of HULL.
      CALL  SUMMATE (X3, XL, TAN_B2, TAN_B1, PH1_HULL_SUM)
      CALL  TRAPEZOID (X3, XL, TAN_B2, TAN_B1,
```

```
     1        PH1_HULL_TRAP)
C      ** PH on MIDDLE SECTION of HULL.
            CALL  SUMMATE (XL, XR, TAN_B_, TAN_B_, PH2_HULL_SUM)
            CALL  TRAPEZOID (XL, XR, TAN_B_, TAN_B_,
     1        PH2_HULL_TRAP)
C      ** PH on RIGHT SECTION of HULL.
            CALL  SUMMATE (XR, X4, TAN_B1, TAN_B2, PH3_HULL_SUM)
            CALL  TRAPEZOID (XR, X4, TAN_B1, TAN_B2,
     1        PH3_HULL_TRAP)
C      ** PH on HULL
            PH_HULL_SUM = PH1_HULL_SUM + PH2_HULL_SUM +
     1        PH3_HULL_SUM
            PH_HULL_TRAP = PH1_HULL_TRAP + PH2_HULL_TRAP +
     1        PH3_HULL_TRAP !
C      *****  TURRET  *****
            XBOT = X2
            WBOT = W2
            XTOP = X6
            WTOP = W6
C      ** POINTS 2 and 6 in U-V COORDINATES.
            U2 = X2 * COS_U + W2 * SIN_U
            V2 = X2 * COS_V + W2 * SIN_V
!
            U6 = X6 * COS_U + W6 * SIN_U
            V6 = X6 * COS_V + W6 * SIN_V
C      ** TURRET PROJECTS over BOTH EDGES of HULL.
            IF (U6 .GT. U2 .AND. V6 .GT. V2) THEN
C        ** DIVIDE TURRET above HULL into THREE SECTIONS
              IF (X2 .LT. X6) THEN
                XL = X2
                XR = X6
                TAN_BOT = TAN_B2
                TAN_TOP = TAN_B1
              ELSE
                XL = X6
                XR = X2
                TAN_BOT = TAN_B1
                TAN_TOP = TAN_B2
              ENDIF
C        ** PH on LEFT SECTION of TURRET
              IF (X7 .LT. XL) THEN
                CALL  SUMMATE (X7,XL,TAN_B1,TAN_B1,
     1            PH1_TURR_SUM)
                CALL  TRAPEZOID (X7,XL,TAN_B1,TAN_B1,
     1            PH1_TURR_TRAP)
              ELSE
                PH1_TURR_SUM = 0.
                PH1_TURR_SUM = 0.
              ENDIF
C        ** PH on MIDDLE SECTION of TURRET
              CALL  SUMMATE (XL,XR,TAN_BOT,TAN_TOP,
     1          PH2_TURR_SUM)
```

```
              CALL  TRAPEZOID (XL,XR,TAN_BOT,TAN_TOP,
     1            PH2_TURR_TRAP)
C           ** PH on RIGHT SECTION of TURRET
              IF (XR .LT. X8) THEN
                  CALL  SUMMATE (XR, X8, TAN_B2, TAN_B2,
     1                PH3_TURR_SUM)
                  CALL  TRAPEZOID (XR, X8, TAN_B2, TAN_B2,
     1                PH3_TURR_TRAP)
              ELSE
                  PH3_TURR_SUM = 0.
                  PH3_TURR_TRAP =0.
              ENDIF

C       ** TURRET PROJECTS over LEFT EDGE of HULL.
          ELSE IF (V6 .GT. V2) THEN
C           ** COMPUTE X INTERSECTION of LEFT EDGE of HULL with
C           ** RIGHT EDGE of TURRET.
              XR = (TAN_B1 * X2 - TAN_B2 * X6 + W6 - W2)
     1             / (TAN_B1 - TAN_B2)
              XR = MIN (XR, X8)
C           ** PH on LEFT SECTION of TURRET
              CALL  SUMMATE (X7,X6,TAN_B1,TAN_B1, PH1_TURR_SUM)
              CALL  TRAPEZOID (X7,X6,TAN_B1,TAN_B1,
     1            PH1_TURR_TRAP)
C           ** PH on RIGHT SECTION of TURRET
              CALL  SUMMATE (X6,XR,TAN_B1,TAN_B2, PH2_TURR_SUM)
              CALL  TRAPEZOID (X6,XR,TAN_B1,TAN_B2,
     1            PH2_TURR_TRAP)
C           **
              PH3_TURR_SUM = 0.
              PH3_TURR_TRAP = 0.

C       ** TURRET PROJECTS over RIGHT EDGE of HULL.
          ELSE IF (U6 .GT. U2) THEN
C       ** COMPUTE X INTERSECTION of RIGHT EDGE of HULL with
C       ** LEFT EDGE OF TURRET
              XL = (TAN_B1 * X6 - TAN_B2 * X2 + W2 - W6)
     1                 / (TAN_B1 - TAN_B2)
              XL = MAX (XL, X7)
C           **
              PH1_TURR_SUM = 0.
              PH1_TURR_TRAP = 0.
C           ** PH on LEFT SECTION of TURRET
              CALL  SUMMATE (XL, X6, TAN_B2, TAN_B1,
     1            PH2_TURR_SUM)
              CALL  TRAPEZOID (XL, X6, TAN_B2, TAN_B1,
     1            PH2_TURR_TRAP)
C           ** PH on RIGHT SECTION of TURRET
              CALL  SUMMATE (X6, X8, TAN_B2, TAN_B2,
     1            PH3_TURR_SUM)
              CALL  TRAPEZOID (X6, X8, TAN_B2, TAN_B2,
     1            PH3_TURR_TRAP)
```

```
!
C     ** TURRET IMAGE IS WITHIN HULL IMAGE
            ELSE
                PH1_TURR_SUM = 0.
                PH1_TURR_TRAP = 0.
                PH2_TURR_SUM = 0.
                PH2_TURR_TRAP = 0.
                PH3_TURR_SUM = 0.
                PH3_TURR_TRAP = 0.
            ENDIF
!
            PH_TURR_SUM = PH1_TURR_SUM + PH2_TURR_SUM +
     1          PH3_TURR_SUM
            PH_TURR_TRAP = PH1_TURR_TRAP + PH2_TURR_TRAP +
     1          PH3_TURR_TRAP !
            WRITE (1,'(A/A/4(A,2F9.4/),A/4(A,2F9.4/),A/A,2F9.4)')
     1      '            SUMMATE  TRAPEZOID',
     1      '    HULL',
     2      ' PH1 =', PH1_HULL_SUM, PH1_HULL_TRAP,
     3      ' PH2 =', PH2_HULL_SUM, PH2_HULL_TRAP,
     4       ' PH3 =', PH3_HULL_SUM, PH3_HULL_TRAP,
     5      ' PH  =', PH_HULL_SUM, PH_HULL_TRAP,
     1      '    TURRET',
     2      ' PH1 =', PH1_TURR_SUM, PH1_TURR_TRAP,
     3      ' PH2 =', PH2_TURR_SUM, PH2_TURR_TRAP,
     4      ' PH3 =', PH3_TURR_SUM, PH3_TURR_TRAP,
     5      ' PH  =', PH_TURR_SUM, PH_TURR_TRAP,
     1      '    TARGET',
     2      ' PH  =', PH_HULL_SUM + PH_TURR_SUM,
     3                PH_HULL_TRAP + PH_TURR_TRAP
        ENDIF
        STOP
        END


        SUBROUTINE  ROTATE (X, Z, XX, ZZ)
        COMMON /ROT/  COS_A, SIN_A
        XX = X * COS_A - Z * SIN_A
        ZZ = X * SIN_A + Z * COS_A
        END


        FUNCTION  DFN (X)
!
C ** FROM HASTINGS APPROXIMATIONS FOR DIGITAL COMPUTERS
!
        F = 0.
        AX = ABS (X)
        IF (AX .GE. 5.) GO TO 1
        F = (((((.5383E-5*AX+.488906E-4)*AX+.380036E-4)*AX +
     1    .0032776263)*AX+.0211410061)*AX+.0498673469)*AX+1.
        F = .5 / ((F**8)**2)
```

```
1       IF (X .GE. 0.) F = 1. - F
        DFN = F
        END

        SUBROUTINE  SUMMATE (X_L, X_R, TAN_1, TAN_2, PH)
!
        DIMENSION  DEVIATE(0:99)
        COMMON /TA/  X_1,W_1,X_2,W_2, XMEAN,XDISP,WMEAN,WDISP
        DATA  (DEVIATE(K), K = 0, 49) /
        *  -2.57583,  -2.17010,  -1.95997,  -1.81191,  -1.69540,
        *  -1.59820,  -1.51411,  -1.43954,  -1.37220,  -1.31059,
        *  -1.25356,  -1.20036,  -1.15035,  -1.10307,  -1.05813,
        *  -1.01522,  -0.97411,  -0.93459,  -0.89648,  -0.85961,
        *  -0.82389,  -0.78919,  -0.75541,  -0.72248,  -0.69032,
        *  -0.65884,  -0.62801,  -0.59776,  -0.56805,  -0.53884,
        *  -0.51008,  -0.48173,  -0.45376,  -0.42616,  -0.39886,
        *  -0.37185,  -0.34513,  -0.31864,  -0.29238,  -0.26630,
        *  -0.24043,  -0.21470,  -0.18913,  -0.16366,  -0.13830,
        *  -0.11304,  -0.08785,  -0.06271,  -0.03761,  -0.01254 /
        DATA  (DEVIATE(K), K = 50, 99) /
        *   0.01254,   0.03761,   0.06271,   0.08785,   0.11304,
        *   0.13830,   0.16366,   0.18913,   0.21470,   0.24043,
        *   0.26630,   0.29238,   0.31864,   0.34513,   0.37185,
        *   0.39886,   0.42616,   0.45376,   0.48173,   0.51008,
        *   0.53884,   0.56805,   0.59776,   0.62801,   0.65884,
        *   0.69032,   0.72248,   0.75541,   0.78919,   0.82389,
        *   0.85961,   0.89648,   0.93459,   0.97411,   1.01522,
        *   1.05813,   1.10307,   1.15035,   1.20036,   1.25356,
        *   1.31059,   1.37220,   1.43954,   1.51411,   1.59820,
        *   1.69540,   1.81191,   1.95997,   2.17010,   2.57583 /
!
C ** Convert coordinates of bottom corner P1 and top corner P2
C ** to standard deviations.
        X1 = (X_1 - XMEAN) / XDISP
        W1 = (W_1 - WMEAN) / WDISP
        X2 = (X_2 - XMEAN) / XDISP
        W2 = (W_2 - WMEAN) / WDISP
C ** Convert tangents of the sides to standard deviation units.
        TAN1 = TAN_1 * XDISP / WDISP
        TAN2 = TAN_2 * XDISP / WDISP
C ** Convert left boundary to standard deviations.
        XL = (X_L - XMEAN) / XDISP
C ** Convert to probability in normal distribution table.
        UL = DFN (XL)
        UL100 = 100. * UL
C ** Convert right boundary to standard deviations.
        XR = (X_R - XMEAN) / XDISP
C ** Convert to probability in normal distribution table.
        UR = DFN (XR)
        UR100 = 100. * UR
C ** I is subscript for inverse normal table 'DEVIATE'.
```

```
C ** DEVIATE(0) = Standard deviations for .005 probability
C ** DEVIATE(1) = Standard deviations for .015 probability
C **         * * * * *
C ** DEVIATE(99) = Standard deviations for .995 probability
        IF (UL100 .NE. 100.) THEN
C     ** Compute deviate nearest left boundary.
            IL = UL100
            X = DEVIATE(IL)
C     ** WB and WT are bottom and top of rectangle given X.
            WB = W1 + TAN1 * (X - X1)
            WT = W2 + TAN2 * (X - X2)
C     ** Compute subscript for 1st probability in "DEVIATE"
C     ** right of left boundary.
            I = UL100 + .5
C     ** Initialize summation by adding or subtracting fractional
C     ** part of beginning increment.
            SUM = (DFN (WT) - DFN (WB)) * (I - UL100)
        ELSE
            SUM = 0.
        ENDIF
C ** Compute subscript for last deviate left of right boundary.
        IMAX = INT (UR100 + .5) - 1
        DO WHILE (I .LE. IMAX)
            X = DEVIATE(I)
            WB = W1 + TAN1 * (X - X1)
            WT = W2 + TAN2 * (X - X2)
            SUM = DFN (WT) - DFN (WB) + SUM
            I = I + 1
        END DO
        IF (UR100 .NE. 100.) THEN
C     ** Compute deviate nearest right boundary.
            IR = UR100
            X = DEVIATE(IR)
C     **  Add or subtract fractional part of last interval.
            WB = W1 + TAN1 * (X - X1)
            WT = W2 + TAN2 * (X - X2)
            SUM = (DFN (WT) -DFN (WB)) * (UR100 -(IMAX+1)) + SUM
        ENDIF
C ** Multiply by interval width [1/100].
        PH = SUM / 100.
        END


        SUBROUTINE  TRAPEZOID (X_L, X_R, TAN_1, TAN_2, PH)

        REAL*8  SUM
        DATA  SQRT_2PI /2.506628/
        COMMON /TA/  X_1,W_1,X_2,W_2, XMEAN,XDISP,WMEAN,WDISP

C ** Convert coordinates of bottom corner P1 and top corner P2
C ** to standard deviations.
        X1 = (X_1 - XMEAN) / XDISP
```

```
      W1 = (W_1 - WMEAN) / WDISP
      X2 = (X_2 - XMEAN) / XDISP
      W2 = (W_2 - WMEAN) / WDISP
C ** Convert tangents of the sides to standard deviation units.
      TAN1 = TAN_1 * XDISP / WDISP
      TAN2 = TAN_2 * XDISP / WDISP
C ** Convert left and right boundaries to standard deviations.
      XL = (X_L - XMEAN) / XDISP
      XR = (X_R - XMEAN) / XDISP
C ** Compute interval width for 10,000 intervals.
      H = (XR - XL) / 10000.
C **  Start trapezoid summation.
C **  WB and WT are bottom and top of rectangle given X.
      WB = W1 + TAN1 * (XL - X1)
      WT = W2 + TAN2 * (XL - X2)
      SUM = EXP (-XL*XL/2.) * (DFN (WT) -DFN (WB)) / 2.
      DO T = 1., 9999.
          X = T * H + XL
          WB = W1 + TAN1 * (X - X1)
          WT = W2 + TAN2 * (X - X2)
          SUM = EXP (-X*X/2.) * (DFN (WT) -DFN (WB)) + SUM
      END DO
      WB = W1 + TAN1 * (XR - X1)
      WT = W2 + TAN2 * (XR - X2)
      SUM = EXP (-XR*XR/2.) * (DFN (WT) - DFN (WB)) / 2. + SUM
      PH = H * SUM / SQRT_2PI
      END
```

APPENDIX C

DISTRIBUTION LIST

	No. Copies
Commander U.S. Army Training and Doctrine Command ATTN: ATCD Fort Monroe, Virginia 23651-5000	1
HQDA Deputy Under Secretary of the Army for Operations Research ATTN: Mr. Walter W. Hollis Room 2E660, The Pentagon Washington, D.C. 20310-0102	1
Deputy Director for Force Structure R&A, J-8 Office of the Joint Chiefs of Staff Room 1E965, The Pentagon Washington, D.C. 20301-5000	1
HQDA Office of the Technical Advisor Deputy Chief of Staff for Operations and Plans ATTN: DAMO-ZD Room 3A538, The Pentagon Washington, D.C. 20310-0401	1
COMMANDER U.S. Army TRADOC Analysis Command ATTN: ATRC Fort Leavenworth, Kansas 66027-5200	1
Director U.S. Army TRADOC Analysis Command-Fort Leavenworth	
ATTN: ATRC-F	1
ATRC-FOQ (Document Control)	1
Fort Leavenworth, Kansas 66027-5200	
Director U.S. Army TRADOC Analysis Command-White Sands Missile Range ATTN: ATRC-W	
ATRC-WSL (Technical Library)	1
ATRC-WFC (Mr. David Durda)	1
White Sands Missile Range, NM 88002-5502	

Director 2
TRADOC Analysis Command
Requirements & Programs Directorate
ATTN: ATRC-RP
Fort Monroe, Virginia 23651-5143

Director 1
U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, Maryland 20814-2797

Director 1
U.S. Army Materiel Systems Analysis Activity
ATTN: AMXSY
Aberdeen Proving Ground, Maryland 21005-5071

U.S. Army Combined Arms Research Library (CARL) 1
ATTN: ATZL-SWS-L
Fort Leavenworth, Kansas 66027

Defense Technical Information Center 2
ATTN: DTIC, FDAC
Building 5, Cameron Station
Alexandria, Virginia 22304-6145

US Army Library 1
Army Study Documentation and Information
Retrieval System (ASDIRS)
ANRAL-RS
ATTN: ASDIRS
Room 1A518, The Pentagon
Washington, D.C. 20310